

Fortranプログラミング入門

-配列(2)-

2.7.1 8 2 8

多次元配列の宣言

☆文法 1) データ型 :: 変数(寸法,寸法,⋯)

2) データ型, dimension(寸法,寸法,⋯) :: 変数

real(8) :: a(4,3)

aは4行3列の配列

a(1,1) : ?	a(1,2) : ?	a(1,3) : ?
a(2,1) : ?	a(2,2) : ?	a(2,3) : ?
a(3,1) : ?	a(3,2) : ?	a(3,3) : ?
a(4,1) : ?	a(4,2) : ?	a(4,3) : ?

2次元配列は
数学の行列
と同じ!!



多次元配列の初期化

real(8) :: a(4,3)

a(:, :) = 0d0

a(1,2) = 1d0

a(2,2:3) = 2d0

a(3:4,:) = 3d0

部分配列を
利用して一気に
初期化!

aは4行3列の配列

a(1,1) : 0d0	a(1,2) : 1d0	a(1,3) : 0d0
a(2,1) : 0d0	a(2,2) : 2d0	a(2,3) : 2d0
a(3,1) : 3d0	a(3,2) : 3d0	a(3,3) : 3d0
a(4,1) : 3d0	a(4,2) : 3d0	a(4,3) : 3d0



多次元配列の初期化

```
real(8) :: a(4,3)
```

```
integer :: i
```

```
a(:, :) = 0d0
```

```
a(1:3,1) = (/1d0,2d0,3d0/)
```

```
a(4,:) = (/i,i=6,10,2/)
```

aは4行3列の配列

a(1,1) : 1d0	a(1,2) : 0d0	a(1,3) : 0d0
a(2,1) : 2d0	a(2,2) : 0d0	a(2,3) : 0d0
a(3,1) : 3d0	a(3,2) : 0d0	a(3,3) : 0d0
a(4,1) : 6d0	a(4,2) : 8d0	a(4,3) : 10d0

1次元配列と
みなし
初期化!



多次元配列の初期化

☆関数 `reshape(一次元配列,次元)`

`real(8) :: a(4,3)`

`a(:, :) = 0d0`

`a(3:4, 1:3) = reshape((/1d0, 2d0, 3d0, 4d0, 5d0, 6d0/), (/2, 3/))`

2行3列の意味



aは4行3列の配列

<code>a(1,1) : 0d0</code>	<code>a(1,2) : 0d0</code>	<code>a(1,3) : 0d0</code>
<code>a(2,1) : 0d0</code>	<code>a(2,2) : 0d0</code>	<code>a(2,3) : 0d0</code>
<code>a(3,1) : 1d0</code>	<code>a(3,2) : 3d0</code>	<code>a(3,3) : 5d0</code>
<code>a(4,1) : 2d0</code>	<code>a(4,2) : 4d0</code>	<code>a(4,3) : 6d0</code>

```
real(8) :: a(4,3), b(6)
```

```
integer :: i
```

```
a(:, :) = 0d0
```

```
b(:) = (/i, i=1,6/)
```

```
a(3:4,1:3) = reshape(b,(/2,3/))
```

aは4行3列の配列

a(1,1) : 0d0	a(1,2) : 0d0	a(1,3) : 0d0
a(2,1) : 0d0	a(2,2) : 0d0	a(2,3) : 0d0
a(3,1) : 1d0	a(3,2) : 3d0	a(3,3) : 5d0
a(4,1) : 2d0	a(4,2) : 4d0	a(4,3) : 6d0

```
real(8) :: a(4,3)
```

```
integer :: i
```

```
a(:, :) = 0d0
```

```
a(3:4, 1:3) = reshape((/(i,i=1,6)/), (/2,3/))
```

aは4行3列の配列

a(1,1) : 0d0	a(1,2) : 0d0	a(1,3) : 0d0
a(2,1) : 0d0	a(2,2) : 0d0	a(2,3) : 0d0
a(3,1) : 1d0	a(3,2) : 3d0	a(3,3) : 5d0
a(4,1) : 2d0	a(4,2) : 4d0	a(4,3) : 6d0

多次元配列の初期化

```
real(8) :: a(4,3)
```

```
integer :: i
```

```
a(:, :) = 0d0
```

```
a(3:4, 1:3) = reshape((/(i,i=1,6)/), (/2,3/))
```

aは4行3列の配列

a(1,1) : 0d0	a(1,2) : 0d0	a(1,3) : 0d0
a(2,1) : 0d0	a(2,2) : 0d0	a(2,3) : 0d0

列優先

多次元配列の出力

2次元配列を単純に出力すると...

```
real(8) :: a(3,2)
```

```
integer :: i
```

```
a(:, :) = reshape((/(i,i=1,6)/), (/3,2/))
```

```
write(*,*) a
```



正解

a(1,1) : 1d0	a(1,2) : 4d0
a(2,1) : 2d0	a(2,2) : 5d0
a(3,1) : 3d0	a(3,2) : 6d0



出力結果

1.00...	2.00...	3.00...
4.00...	5.00...	6.00...

多次元配列の出力

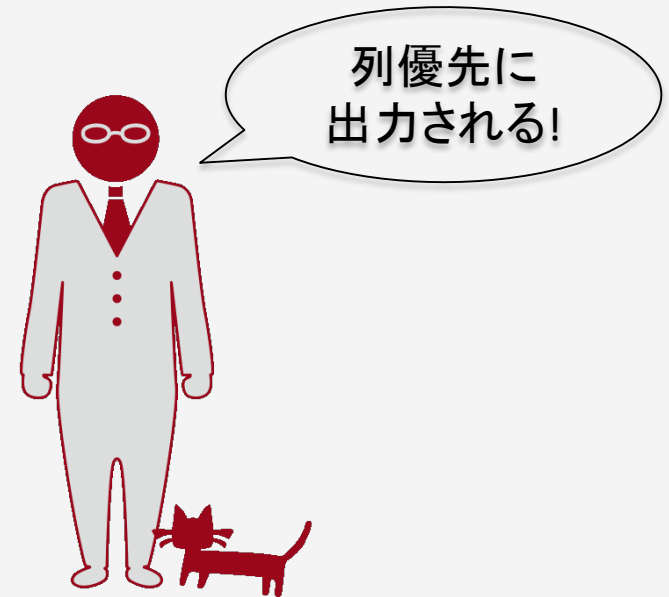
2次元配列を単純に出力すると...

```
real(8) :: a(3,2)
```

```
integer :: i
```

```
a(:, :) = reshape((/(i,i=1,6)/), (/3,2/))
```

```
write(*,*) a
```



正解

a(1,1): 1d0	a(1,2): 4d0
a(2,1): 2d0	a(2,2): 5d0
a(3,1): 3d0	a(3,2): 6d0



出力結果

1.00...	2.00...	3.00...
4.00...	5.00...	6.00...

多次元配列の出力

2次元配列を単純に出力すると...

```
real(8) :: a(3,2)
```

```
integer :: i
```

```
a(:, :) = reshape((/(i,i=1,6)/), (/3,2/))
```

```
write(*,100) (a(i,:),i=1,3)
```

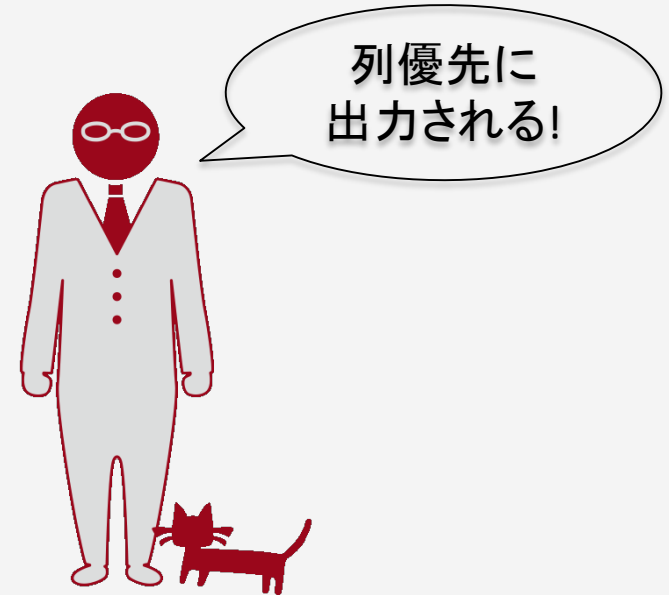
```
100 format(2f5.2)
```

正解

列数

行数

a(1,1): 1d0	a(1,2): 4d0
a(2,1): 2d0	a(2,2): 5d0
a(3,1): 3d0	a(3,2): 6d0



出力結果

1.00	4.00
2.00	5.00
3.00	6.00

5 × 4の単精度実数型の変数Aを宣言しろ. Aの行番号が奇数で列番号が偶数の要素に1を代入せよ. それ以外には0を代入せよ. 但し, 出力は4f5.2を用いて正しく表示せよ.

答え1

```
program exam1
  implicit none
  real :: A(5,4)
  integer :: i
  A(:, :) = 0e0
  A(1:5:2, 2:4:2) = 1e0
  write(*, 100) (A(i, :), i=1, 5)
  stop
100 format(4f5.2)
end program exam1
```

答え2

```
program exam1
  implicit none
  real :: A(5,4)
  integer :: i, j
  A(:, :) = 0e0
  do i=1, 5, 2
    do j=2, 4, 2
      A(i, j) = 1e0
    end do
  end do
  write(*, 100) (A(i, :), i=1, 5)
  stop
100 format(4f5.2)
end program exam1
```

☆文法 where(論理型の配列)

論理型配列で真の要素に対して実行する文

end where

例えば...

```
real(8) :: A(3,2)
```

```
A(:, :) = 0d0
```

```
A(1:3:2, 2) = 1d0
```

この時点では

```
where (A==0)
```

```
  A = -5d0
```

```
end where
```



a(1,1) : 0d0	a(1,2) : 1d0
a(2,1) : 0d0	a(2,2) : 0d0
a(3,1) : 0d0	a(3,2) : 1d0

☆文法 where(論理型の配列)

論理型配列で真の要素に対して実行する文

end where

例えば...

```
real(8) :: A(3,2)
```

```
A(:, :) = 0d0
```

```
A(1:3:2, 2) = 1d0
```

```
where (A==0)
```

```
  A = -5d0
```

```
end where
```

論理型配列 $A==0$ は...



.true.	.false.
.true.	.true.
.true.	.false.

☆文法 where(論理型の配列)

論理型配列で真の要素に対して実行する文

end where

例えば...

real(8) :: A(3,2)

A(:, :) = 0d0

A(1:3:2,2) = 1d0

where (A==0)

A = -5d0

end where

論理型配列 A==0 は...

.true.	.false.
.true.	.true.
.true.	.false.

.true.のみ実行され, Aは...

a(1,1) : -5d0	a(1,2) : 1d0
a(2,1) : -5d0	a(2,2) : -5d0
a(3,1) : -5d0	a(3,2) : 1d0

配列同士の演算:

1次元配列と同じで, 成分ごとの演算!!

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \circ \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{pmatrix} = \begin{pmatrix} a_{11} \circ b_{11} & a_{12} \circ b_{12} & a_{13} \circ \cdots & a_{1m} \circ b_{1m} \\ a_{21} \circ b_{21} & a_{22} \circ b_{22} & a_{23} \circ \cdots & a_{2m} \circ b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} \circ b_{n1} & a_{n2} \circ b_{n2} & a_{n3} \circ \cdots & a_{nm} \circ b_{nm} \end{pmatrix}$$

$$a \circ \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{pmatrix} = \begin{pmatrix} a \circ b_{11} & a \circ b_{12} & a \circ \cdots & a \circ b_{1m} \\ a \circ b_{21} & a \circ b_{22} & a \circ \cdots & a \circ b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a \circ b_{n1} & a \circ b_{n2} & a \circ \cdots & a \circ b_{nm} \end{pmatrix}$$

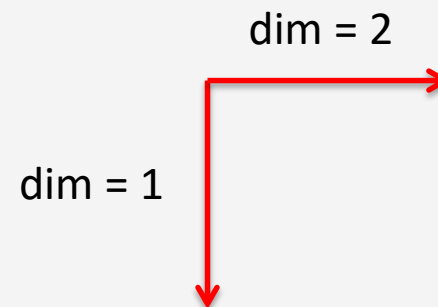
$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \circ b = \begin{pmatrix} a_{11} \circ b & a_{12} \circ b & a_{13} \circ \cdots & a_{1m} \circ b \\ a_{21} \circ b & a_{22} \circ b & a_{23} \circ \cdots & a_{2m} \circ b \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} \circ b & a_{n2} \circ b & a_{n3} \circ \cdots & a_{nm} \circ b \end{pmatrix}$$

但し $\circ$ は $+$, $-$, $*$, $/$, $**$ のいずれかである.

関数名	意味	入力引数の型
<code>sum(A,dim)</code>	dim番目の次元に対する総和	Aはn次元配列 単/倍精度実数 単精度複素数
<code>maxval(A,dim)</code>	dim番目の次元の最大値	Aはn次元配列 整数 単/倍精度実数
<code>minval(A,dim)</code>	dim番目の次元の最小値	Aはn次元配列 整数 単/倍精度実数
<code>size(A,dim)</code>	dim番目の次元の寸法	Aはn次元配列

例えば

$$A = \begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix}$$



に対して

$$\text{maxval}(A, 1) = (4 \quad 8 \quad 12)$$

$$\text{sum}(A, 1) = (10 \quad 26 \quad 42)$$

$$\text{maxval}(A, 2) = \begin{pmatrix} 9 \\ 10 \\ 11 \\ 12 \end{pmatrix}$$

$$\text{sum}(A, 2) = \begin{pmatrix} 15 \\ 18 \\ 21 \\ 24 \end{pmatrix}$$

関数名	意味	入力引数の型
matmul(A,B)	AとBの行列積 (戻り値の配列の寸法 に注意)	A,Bは2次元配列 単/倍精度実数 単精度複素数
transpose(A)	Aの転置行列 (A(m,n)のとき, 戻り値 の配列寸法は(n,m))	Aは2次元配列 整数 単/倍精度実数